Resilient and Robust Strategies for Swarming Ad-hoc Networks

Evelyne Akopyan, Emmanuel Lochin, Riadh Dhaou, Bernard Pontet, Jacques Sombrin

Enhancing the Reliability of Swarming Ad-hoc Networks

Evelyne Akopyan[1,3*], Emmanuel Lochin[2†], Riadh Dhaou[3†], Bernard Pontet[4†], Jacques Sombrin[1†]

[1*]TéSA, Toulouse, France.
[2*]Fédération ENAC ISAE-SUPAERO ONERA, Université de Toulouse, France.
[3*]IRIT, Université de Toulouse, CNRS, Toulouse INP, France.
[4*]CNES, Toulouse, France.

*Corresponding author(s). E-mail(s): evelyne.akopyan@tesa.prd.fr;
Contributing authors: emmanuel.lochin@enac.fr; riadh.dhaou@irit.fr;
bernard.pontet@cnes.fr; jacques.sombrin@tesa.prd.fr;
†These authors contributed equally to this work.

Abstract

We investigate the evolution of the reliability of a mobile communication network, *i.e.*, its capacity to avoid and withstand faults (called *robustness*) and to maintain proper functioning if faults occur nevertheless (called *resilience*). Our case study focuses on a swarm of nanosatellites orbiting the Moon and operating as a distributed space interferometer. The objective of this study is to evaluate the impact of graph division techniques on the robustness and resilience of the system, simultaneously. A high reliability level is key to guaranteeing proper quality of service by recovering from impairments while preserving the primary function or mission of the mobile network. By analyzing the effects of exploration and random selection algorithms on the network reliability, our simulations indicate that fair graph division consistently strengthens the robustness of swarming ad-hoc networks, regardless of the algorithm employed. In addition, our analysis highlights the superior performance of sequential exploration algorithms, such as MIRW, in optimizing robustness while preserving a decent level of resilience.

Keywords: Mobile Ad-hoc Networks, Architecture, Robustness, Resilience, Graph Division

1 Introduction

Enhancing the reliability of complex distributed systems and ad-hoc networks has been a central research area for many years. Such systems must be robust and resilient against faults to guarantee a satisfactory quality of service. System robustness is its ability to avoid or withstand internal and external faults: it usually regroups prevention mechanisms. For instance, a drone fleet is deemed robust if it incorporates an energy management mechanism that ensures a reduction in energy consumption, extends the system lifespan, and minimizes the risk of failure due to energy depletion ([1]). Resilience is a distinct concept that characterizes a system's ability to maintain functioning despite the presence of faults, and can be assimilated to fault tolerance. For example, cellular networks demonstrate resilience by deploying redundant equipment to address network faults ([2]). Although adding redundancy is a common strategy to enhance resilience, ad-hoc networks and their characteristic lack of infrastructure cannot rely on external equipment to improve fault tolerance. Consequently, the design of these systems must consider both robustness and resilience constraints.

This analysis is particularly essential for Distributed Space Systems (DSS) as their decentralized architecture and reliance on inter-node communication make them highly sensitive to both robustness and resilience challenges, necessitating a comprehensive evaluation that simultaneously considers these interdependent factors.

However, most studies related to the reliability of complex systems are either focused on robustness, or resilience, but rarely both. This approach is problematic because these concepts can be antagonistic by definition. If we consider an ad-hoc network, a robustness mechanism would be to optimize packet routing, *i.e.*, reduce the number of packet transmissions to the bare minimum. On the other hand, a resilience mechanism would be to add redundancy and send packets multiple times, in case there is loss on the network. This simple example illustrates the existing tradeoff between robustness and resilience. This study aims to thoroughly analyze the reliability of swarming ad-hoc networks by addressing both robustness and resilience, which is lacking in the literature.

As mentioned above, our study focuses on a scenario involving a nanosatellite swarm in orbit around the Moon that performs as a distributed radio telescope in outer space ([3]). This system operates as a Mobile Ad-hoc Network (MANET), and its configuration as a space observatory raises several communication challenges. First, the network lacks infrastructure and relies exclusively on wireless Inter-Satellite Links (ISL), with nodes serving as sources, destinations, and routers. In addition, the swarm operates as a Distributed Space System (DSS): unlike conventional observation telescopes, which carry one or many measuring devices on a single satellite, a swarm of nanosatellites forms a single instrument distributed across many satellites. The distributed nature of this system imposes a significant constraint on the data transmission model, in addition to the lack of infrastructure: each node is required to share its data with all other nodes in a manner that resembles any-to-any communication. This data-sharing process is essential for space observatories to generate a global, comprehensive image of space observations, which is subsequently transmitted to a base station on Earth.

Nevertheless, the data generated from space observations can be extensive, reaching several gigabits per nanosatellite. The simultaneous transmission of these data packets may result in ISL congestion, potentially leading to packet loss. Each lost packet degrades the accuracy of the global image, resulting in reduced information transmitted to the base station. Moreover, data transmission poses an energy challenge on the nanosatellite scale: if the nanosatellite distribution within the swarm is heterogeneous, the central nanosatellites will more likely transmit a higher volume of packets, which will consume their energy faster and result in failure by energy depletion, thus causing a consistent decline in accuracy for the global image. Consequently, a distributed MANET must distribute the data load fairly among each network node. In a previous study ([4]), we proposed an approach based on fair graph division of the swarm to balance the network load among nanosatellites. Dividing the swarm into multiple groups of nanosatellites significantly alleviates the overall energy consumption of the swarm, without impacting the functioning of the system. However, the study of network load alone is not sufficient to assess the reliability of a swarming ad-hoc network because it is based solely on a robustness metric and cannot thoroughly describe the system's reaction to faults.

Therefore, the primary objective of this follow-up study is to provide an extensive analysis of network reliability under both robustness and resilience constraints, which is rare in the MANET and DSS literature. Our main contribution lies in emphasizing the beneficial influence of fair graph division mechanisms on the robustness and resilience of a distributed MANET, building upon our prior investigations. We introduce a methodology for evaluating robustness and resilience levels before and after graph division, shedding light on any positive or negative effects that graph division may have on network reliability. Enhancing these levels is crucial for improving the system's reaction to faults and ensuring the proper functioning of the DSS. Our findings reveal that fair graph division significantly enhances the system's robustness, but it also highlights the tradeoff aforementioned: while specific resilience metrics may experience improvement, it comes at the cost of others.

The remainder of this paper is organized as follows: Sec. 2 presents a literature review of related work, and Sec. 3 introduces the graph division mechanism and the division algorithms to be evaluated. The methodology used to assess robustness and resilience levels in a distributed MANET is presented in Sec. 4. The impact analysis of graph division and the nanosatellite swarm scenario results are detailed in Sec. 5. Finally, Sec. 6 summarizes our study and discusses future research on the subject.

2 Related work on system reliability

This section presents a compilation of studies relevant to the proposed investigation. We have categorized this state-of-the-art review into two parts: the first focuses on robustness, and the second on resilience.

2.1 Robustness

The robustness of a network characterizes its ability to maintain functionality and withstand internal and external faults. Achieving a high level of robustness is essential for countering intentional attacks and ensuring operation in hostile environments.

Energy efficiency is a crucial metric for robustness, particularly in Wireless Sensor Networks (WSN), where routing protocols play a vital role in finding optimal routes while considering energy consumption. Hierarchical protocols such as Low Energy Adaptive Clustering Hierarchy (LEACH) enhance the overall efficiency of WSNs, increasing network lifetime. A comparative analysis by [5] evaluates 27 routing protocols derived from LEACH, considering metrics such as scalability, energy efficiency, and communication method. Although each algorithm has its strengths and weaknesses, they collectively contribute to improved network load balancing and reduce energy consumption, enhancing overall robustness.

Identifying critical nodes is another crucial aspect of building a robust network. [6] propose an algorithm based on weighted betweenness centrality to pinpoint central nodes in complex transportation networks. Identifying these major nodes is crucial for network safety and reliability. The authors demonstrate the accuracy of the algorithm in identifying critical nodes efficiently, with results consistent with real-world scenarios.

Examining the impact of attacks on MANETs, [7] focus on the targeted attack scenario in which the highest-centrality nodes are affected in dynamic graphs of varying density levels. The authors establish betweenness centrality as a precise measure of node importance in highly connected graphs. However, in sparsely connected graphs, node degree and eigenvector centrality emerge as more accurate metrics for node significance.

In the context of robust networks, ensuring a strong Quality of Service (QoS) is paramount. [8] propose the Reliable and Stable Topological Change Adaptive Ad-hoc On-demand Multipath Distance Vector (RSTA-AOMDV) routing protocol to maintain QoS during data transmission. This protocol considers the local information of nodes while forwarding packets, resulting in superior performance in terms of Packet Delivery Ratio (PDR), throughput, and packet transmission time compared to traditional routing protocols such as AOMDV-MCA or SR-MQMR. While robustness focuses on preventing failures, resilience addresses the ability to recover from them, making both aspects interdependent.

2.2 Resilience

The resilience of a network refers to its ability to recover from faults when they occur. The extensive literature on system resilience and sustainability spans various domains, including communication networks (referenced in [2]), transportation networks (referenced in [9]), environmental sustainability (referenced in [10]), and socio-ecological systems (referenced in [11]). Despite differences across these domains, the definitions of resilience and the measurement methods remain consistent.

The Internet serves as an example of a communication network with low resilience. [12] introduce an architectural framework that enhances communication network resilience by incorporating resilience strategies and principles into the analysis. The

authors underscore the importance of considering resilience in communication network design.

In complex networks, [13] present metrics for evaluating resilience, particularly in resource trade networks. Their evaluation involves statistically characterizing resilience through redundancy (the number of duplicate elements), diversity (variety, distribution, and disparity of elements), and modularity (a system's capacity to fragment into distinct communities to contain shocks or stress).

For MANETs, [14] propose the Multipath-ChaMeLeon (M-CML) resilient routing protocol. M-CML, based on a multipath approach, reduces end-to-end delay and improves the PDR compared to the Optimized Link State Protocol (OLSR), albeit with a higher number of redundant packets.

Resilience in MANETs also plays a crucial role in disaster management. [15] introduce a hybrid Wireless Multi-Hop Network (WMHN) architecture, combining Flying Mesh Networks (FMN) and opportunistic MANETs to enhance connectivity with survivors, reduce packet loss, and decrease end-to-end delay.

To evaluate the resilience of public transportation networks, [16] assesses the number and availability of alternative paths. A resilient transportation network, defined by extensive route redundancy, is characterized by efficient and reasonably short alternative paths that consider travel costs. In addition, the authors evaluate the connectivity and accessibility of the studied metro networks.

Our position in this literature review is to analyze the impact of a given mechanism and/or algorithm on both robustness and resilience, simultaneously. We consider that these two properties should not be studied separately, because they have a mutual influence on each other. Hence, we propose a general methodology to combine the assessment of robustness and resilience levels in MANETs. This methodology can be a great help for researchers conducting comprehensive studies of robustness and resilience levels in complex systems.

3 Graph division

The main objective of this paper is to highlight the importance of fair graph division for robustness and resilience improvement in MANETs. Graph division is a mechanism that assigns each vertex of the graph to a vertex set called *group*. Note that the edges remain unchanged; therefore, the groups are still connected with each other and not physically separated. Graph division is a particularly interesting solution for reducing the number of routed packets in MANETs, in the same way that graph clustering helps reduce the number of routed packets in Wireless Sensor Networks ([5]). Graph division has a different purpose from graph clustering: the core mechanism of graph division is to obtain groups that are similar to the original graph, whereas the core mechanism of graph clustering is to obtain groups such that the vertices within a given group are similar to each other.

We talk about fair graph division when the obtained groups are similar to the original graph and similar to each other: the group size is a straightforward fairness metric, but other metrics can be chosen. The evaluation of fairness in graph division has been thoroughly studied by the authors in a previous paper ([4]).

3.1 Notations and metrics adaptation

Ad-hoc networks can be efficiently represented by graphs ([17], [18]). Throughout this paper, we model any MANET as a temporal graph $G_t = (V_t, E_t)$, with a vertex set V_t of size $|V_t| = n(t)$ representing the network nodes (*e.g.* the nanosatellites), and an edge set E_t of size $|E_t| = m(t)$ representing the communication links (*e.g.* the ISL). Each edge has a weight representing the communication cost. In our case study, the cost of using a specific ISL depends on the distance between two nanosatellites. For simplicity, we assume that the communications are performed over symmetrical duplex links, which makes G_t an undirected weighted graph.

We consider a MANET scenario in which each member of the network needs to share its data packet (*e.g.* interferometry measurements) with all the other members, and then combine the data to obtain a global cross-correlation matrix of data collected by the network. This means that each member of the network should communicate with the others. Mathematically speaking, we denote $N_G(t)$ as the total number of vertex pairs that need to communicate in an undivided graph, called *flow number*, and defined as follows:

$$N_G(t) = \frac{n(t) \times (n(t) - 1)}{2} \tag{1}$$

The robustness and resilience metrics described in Sec. 4 depend on the flow number $N_G(t)$ in the graph at a given time. However, after performing graph division, the number of source-destination pairs is altered, because the vertices only need to communicate with the vertices of the same group as a first step. By limiting data transmission to intra-group transmissions only, each group computes a partial cross-correlation data matrix. Once all partial matrices are computed, they are shared among the groups and recomputed locally with the additional data to finally obtain the global cross-correlation data matrix. In other words, the flow number of a divided graph is the sum of intra-group and inter-group transmissions.

We denote $G_t^* = (V_t^*, E_t)$ the graph obtained after graph division. The edge set remains unchanged, but the vertices are assigned to x groups in V_t^*:

$$V_t^* = \{V_t^0, V_t^1, ..., V_t^{x-1}\} \text{ and } \bigcup_{i<x} V_t^i = V_t$$

Therefore, the flow number of the undirected, weighted, divided graph G_t^* becomes:

$$N_G^*(t) = \sum_{i<x} \frac{n_i(t) \times (n_i(t) - 1)}{2} + x(x-1) \tag{2}$$

where the sum represents the number of intra-group transmissions, and the product represents the number of inter-group transmissions. Thus, the robustness and resilience metrics defined in Sec. 4 are to be adapted to consider only the new source-destination pairs, and not the entire graph.

3.2 Graph division algorithms

Graph division algorithms are often inspired by sampling algorithms ([19]). We focus on two main families of algorithms: random selection algorithms and exploration

algorithms. The algorithms listed below and their implementation are thoroughly explained in [20].

Random selection algorithms are based on the random selection of vertices and/or edges to create a graph sample. In graph division, a random selection algorithm randomly assigns a vertex and/or edge to a group. We choose to work with the Random Node Division algorithm (RND), which performs random selection on vertices only. We decided not to work with edge-related algorithms such as Random Edge Division (RED) because our graph is not guaranteed to be connected; hence, some vertices might never be assigned to any group.

Exploration algorithms are based on neighborhood discovery of a vertex to select the members of a given group. The choice among the neighborhood can be random, or have a certain probability depending on the characteristics of the vertex. A simple example of exploration algorithms is those based on random walks. We choose to work with the Multiple Independent Random Walks algorithm (MIRW), which, as the name suggests, runs several random walks in parallel in a graph. The particularity of MIRW is that it performs a random jump in the graph whenever a random walk is stuck (*i.e.*, there are no more free vertices in the neighborhood), to continue the random walk elsewhere in the graph: this mechanism ensures that each vertex in the graph belongs to a group by the end of the division process. Another example of an exploration algorithm is the Forest Fire Division algorithm (FFD), which selects each vertex in a neighborhood with a given probability p. Similarly, FFD performs a random jump in the graph if there are no more available vertices in a neighborhood.

In the following sections, we will also study the clustering algorithm K-means for comparison: this algorithm is well-known and widely used in the clustering community, although it is not designed to create similar groups, but rather regroup similar nodes together. Still, the comparison with division algorithms is necessary to fully grasp the interest of each technique.

4 Assessment of robustness and resilience levels

We propose a graph theory-based methodology, applicable to divided and undivided graphs, to assess the levels of robustness and resilience in a network. In this section, we introduce six graph theory metrics that are used to characterize the levels of robustness and resilience.

4.1 Definition of robustness metrics

We focus on three metrics to evaluate the level of robustness of the network: the flow robustness, routing cost and network efficiency. These metrics have been chosen because, when combined, they accurately describe the dynamics of packet transmission in a network.

Flow robustness, denoted $F_t(G)$: the flow robustness of a divided or undivided graph estimates the proportion of the graph that can be reached by a packet through multi-hop. In graph theory terms, the flow robustness of a graph G at time t is the proportion of vertex pairs in G_t connected by a path, and is defined as:

$$F_t(G) = \frac{f_t(G)}{N_G(t)} \tag{3}$$

where $f_t(G)$ is the number of pairs effectively connected by a path at time t, and $N_G(t)$ is the flow number of the graph defined by Eq. 1 (respectively Eq. 2 for a divided graph). When the flow robustness equals one, the graph is connected, *i.e.*, each vertex can reach any other vertex in the graph. Otherwise, the graph is disconnected, so the packets cannot be spread across the entire graph.

Routing cost, denoted $R_t(G)$: the routing cost is the total number of packets to be transmitted through the network, including multi-hop retransmissions, in order to reach all the destination vertices from all the source vertices. We assume that the packets always take the shortest path between the source and destination vertices. This assumption is realistic in a multicast routing scenario, where the first instance of the packet to reach the destination would have travelled the shortest path. In this study, as described in Sec. 3.1, all vertices should share their data packets with all other vertices of the network: this means that the vertices in the graph act as sources and destinations. The routing cost of a graph G at time t can be simply defined as:

$$R_t(G) = \sum_{u,v \in V_t^2} d(u,v) \tag{4}$$

where $d(u,v)$ is the shortest path length between the vertices u and v, also called distance between u and v. The routing cost must be as small as possible because any node in a MANET consumes energy when transmitting a packet. Thus, by transmitting fewer packets in the network, the nodes save more energy and can last longer.

Network efficiency, denoted $\Theta_t(G)$: the network efficiency is a coefficient between 0 and 1 that characterizes the distances between each vertex pair in the graph: if the distances are globally short, then the network efficiency is high. This means that the packets will need to perform fewer hops and thus be retransmitted less often, eventually saving energy for the nodes of the network. To evaluate the network efficiency, we first compute the pair efficiency for each vertex pair in G_t, denoted $\theta(u,v)$:

$$\theta(u,v) = \frac{1}{d(u,v)}$$

The pair efficiency of vertices u and v equals 0 if there is no path connecting these vertices, and 1 if they are adjacent. The network efficiency of a graph G at time t is then simply defined as the average pair efficiency in G_t:

$$\Theta_t(G) = \frac{1}{2N_G(t)} \sum_{u,v \in V_t^2, u \neq v} \theta(u,v) \tag{5}$$

In a divided graph G_t^*, the network efficiency only accounts for the source-destination pairs defined by the groups and is written as:

$$\Theta_t(G^*) = \frac{1}{2N_G^*(t)} \sum_{V_t^i \in V_t^*} \left(\sum_{u,v \in (V_t^i)^2} \theta(u,v) \right) \tag{6}$$

Either way, in a robust system, the network efficiency should tend to 1 as it implies that fewer packet retransmissions are needed for multi-hop.

4.2 Resilience

As stated in Sec. 2.2, the resilience can be characterized by various metrics. We propose the following three metrics to evaluate the level of resilience in the network: path redundancy, path disparity, and node criticity. We selected these metrics because they thoroughly evaluate the reliability of the paths taken by the packets.

Path redundancy, denoted $\Psi_t(G)$: the path redundancy is a measure of the number of alternative paths in the graph. A path is considered an alternative if it passes through at least one vertex different from the reference path without being consistently longer than the shortest path. A great path redundancy is proof of a strong level of resilience, because the packets can be routed from source to destination through different vertices, should a vertex go down. Path redundancy is defined as the average number of alternative paths between each two vertices in G_t:

$$\Psi_t(G) = \frac{1}{2N_G(t)} \sum_{u,v \in V_t^2} |P_{uv}| \tag{7}$$

where P_{uv} is the set of paths connecting vertices u and v. If there is no path between vertices u and v, the path redundancy of this pair equals 0.

Path disparity, denoted $\Delta_t(G)$: the path disparity is complementary to path redundancy. It is a measure of the difference between the alternative paths, *i.e.*, the number of different vertices composing the paths. A high path disparity ensures that in the presence of a fault on a path, a packet can transit on a completely separate path and hence not be impacted by the fault. We first compute, for each vertex pair, the proportion of different vertices in the alternative paths available between the two vertices, which is called pair disparity:

$$\delta(u,v) = \frac{1}{2|P_{uv}|} \sum_{p_i,p_j \in P_{uv}^2} \frac{|p_i(u,v) \oplus p_j(u,v)|}{d(u,v) - 1}$$

where the numerator represents the number of distinct vertices between the two alternative paths p_i and p_j. The pair disparity is maximal if the paths are vertex-disjoint, *i.e.*, they share no common vertex (apart from source and destination), and is minimal if the paths are identical, or if there is at most one path between two vertices. Fig. 1 illustrates the computation of pair disparity. The path disparity of a graph G at time t is then easily derived as the average pair disparity of the graph:

$$\Delta_t(G) = \frac{1}{2N_G(t)} \sum_{u,v \in V_t^2} \delta(u,v) \tag{8}$$

Node criticity: the node criticity is an indicator of the presence and number of critical vertices in a graph. A vertex is considered critical if a significant proportion of traffic passes through it, *i.e.*, if it is present in a large proportion of shortest paths in the graph. Thus, node criticity is usually based on a centrality measure of the vertices,

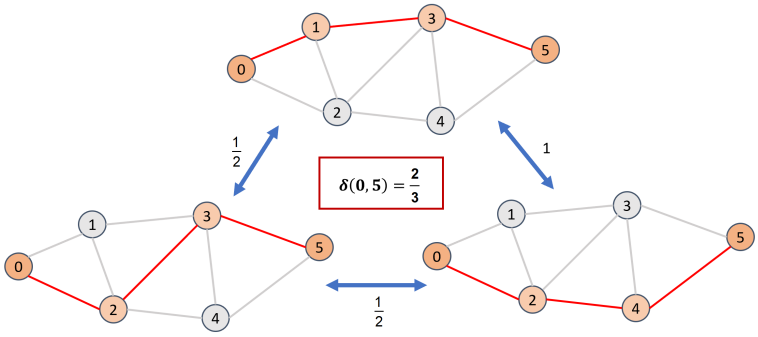


Fig. 1: Illustration of pair disparity computation: the pair disparity between vertices 0 and 5 equals $\frac{2}{3}$, or approximately 67%.

such as degree centrality or betweenness centrality ([6]). We choose the normalized betweenness centrality as our criticity measure, defined for a vertex i as follows:

$$BC_t(i) = \frac{1}{2N_G(t)} \sum_{u,v \in V_t^2} \frac{|P_{uv}(i)|}{|P_{uv}|} \tag{9}$$

where $P_{uv}(i)$ is the set of paths from u to v passing through vertex i. The distribution of betweenness centrality values is sufficient to describe node criticity; however, the analysis of the number of extreme values can be a great addition to grasp the dynamics of the graph. As such, we define the critical set $C_t(G)$ of a graph as the subset of vertices whose betweenness centrality is higher than a threshold value ϵ at a given time t:

$$C_t(G) = \{i \in V_t \,|\, BC_t(i) \geq \epsilon\} \tag{10}$$

By definition, the critical set contains the most critical vertices of a graph, *i.e.*, the vertices that are the most vulnerable to faults. A resilient network tends to have the smallest critical set possible.

To summarize, we have presented three metrics for robustness assessment (flow robustness, routing cost and network efficiency) and three metrics for resilience assessment (path redundancy, path disparity and node criticality). In the following section, we demonstrate the relevance of this selection of metrics to thoroughly characterize the dynamics of packet transmission and the reliability of the available paths in the network.

5 Impact analysis of graph division: application to nanosatellite swarms

In this section, we present a practical application of the theoretical approach described above. We focus on a nanosatellite swarm deployed in orbit around the Moon to perform space observations by analyzing very low frequencies. We first describe the

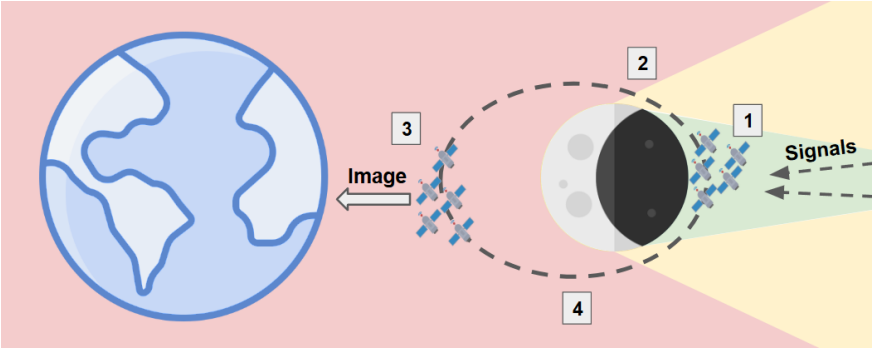


Fig. 2: Illustration of the four stages of operation of the swarm: observation (1), computation (2), uplink (3) and idle (4).

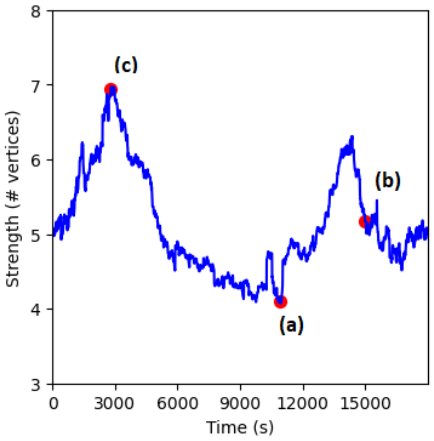
functioning of the system in Sec. 5.1 as well as the dataset used in Sec. 5.2. Finally, we analyse the impact of graph division on system reliability in Sec. 5.3 and prove that fair graph division makes a non-trivial improvement in the robustness and resilience of this system.

5.1 Context of the operation

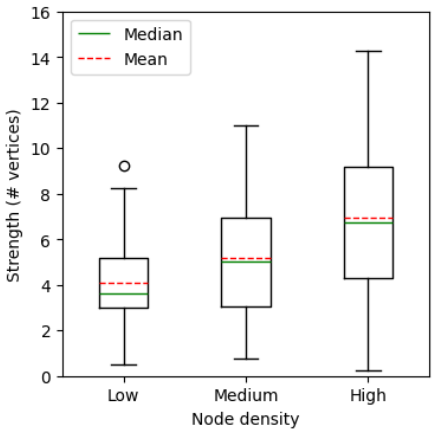
The study of Dark Age signals is a key research topic for understanding the early days of the universe and the formation of stars ([21]). These signals consist of very low-frequency interferences, typically below 10 MHz. To this day, low-frequency interferometers are deployed on the surface of the Earth, in extreme weather conditions for better observation (*e.g.* the interferometers ALMA and VLTI, both deployed in the Atacama desert in Chile). Unfortunately, low-frequency signals are still hardly observable by these interferometers because of ionospheric distortion and Radio Frequency Interferences (RFI). One solution to alleviate this issue is to deploy a swarm of nanosatellites directly into space, as presented in the Nanosatellites for a Radio Interferometer Observatory in Space study (NOIRE) carried out by [3]. This study proves that the deployment of a nanosatellite swarm in orbit around the Moon is a viable solution for space observations of Dark Age signals, as the Moon naturally shields the swarm from Earth's RFI when the swarm flies over the dark side of the Moon.

To be considered a valid space radiotelescope, the nanosatellite swarm must operate similarly to ground-based telescopes. There are four main stages in the process, as depicted in Fig. 2:

1. Observation phase: while on the dark side of the Moon, *i.e.*, when the swarm is shielded from the RFI, each nanosatellite collects raw observation data of low frequencies from space;
2. Computation phase: each nanosatellite shares its data with each other member of the swarm and receives their data in return, then computes a cross-correlation matrix from all available data to produce a global interferometry image of space;
3. Uplink phase: the swarm finally uplinks the image to a base station on Earth, preferably when the swarm-to-Earth distance is minimal.



(a) Evolution of average vertex strength over one revolution



(b) Strength distributions at three levels of swarm density: low (a), medium (b), and high (c)

Fig. 3: Estimation of swarm density: analysis of vertex strength

4. Idle phase: the nanosatellites restore their energy, and the swarm can spatially reconfigure if requested by the base station.

It is primordial for such a system to be reliable, *i.e.*, both robust and resilient, to perform its mission for the longest time possible.

5.2 Description of the dataset

We use nanosatellite position data generated by a trajectory simulator developed for the French National Institute of Space Studies (CNES, [22]). We consider a swarm of 50 nanosatellites orbiting the Moon. The dataset provides the positions of each nanosatellite of the swarm and their trajectory over time. There is no bootstrap stage, so the behavior of the swarm is stable over time.

The data consist of the (x, y, z) coordinates of each nanosatellite in the Moon-centered coordinate system, sampled every 10 seconds over a complete duration of 24 hours. The swarm performs a revolution around the Moon in 5 hours. The behavior of the swarm is quasi-identical for each revolution, so we choose to analyze only one period of revolution for simplicity.

During a revolution, the topology of the swarm evolves, particularly in terms of nanosatellite density. Fig. 3a shows the temporal evolution of the average vertex strength of the swarm, which is equivalent to the vertex degree for weighted graphs. We can easily see that the swarm density is time-dependent, as the average vertex strength is not constant in time. Fig. 3b displays a close-up of vertex strength distribution in the swarm, at three typical levels of density: a low-density topology, represented by a low average strength (point a), a medium-density topology, represented by a medium

average strength (point b), and a high-density topology, represented by a high average strength (point c). We can see that the statistical range increases with the level of density, meaning that the strength distribution is particularly heterogeneous in dense topologies. This also indicates that the swarm density is very heterogeneous, which implies that some nodes located in dense areas can be overused and consume their energy faster than the average.

The evaluation protocol of robustness and resilience metrics, presented below, is performed over the complete revolution as well as the three topology close-ups to fully grasp the dynamics of robustness and resilience metrics in the swarm.

5.3 Simulation results

To understand the influence of fair graph division on the robustness and resilience of the network, we first compute the robustness metrics (flow robustness, routing cost and network efficiency) and resilience metrics (path redundancy, path disparity and node criticity) on the original undivided graph. To do so, we compute these metrics at each timestamp over a complete revolution around the Moon. These results represent the reference performance of the network in terms of robustness and resilience.

Then, we partition the graph into groups with the chosen algorithms (RND, MIRW, FFD or k-means) in the initial phase of the swarm (*i.e.*, at timestamp 0), and fix the groups for the rest of the simulation so that the vertices cannot change groups during the simulation. The results presented below are based on a graph division into 10 groups, because it is the optimal number of groups to minimize the overall energy consumption of the swarm (see previous studies in [4] and [20]). We proceed to compute robustness and resilience metrics on the divided graph, similarly to the original graph. Because the proposed algorithms are based on random mechanisms, this operation (division and computation) is independently repeated 30 times to obtain a good estimation of the behavior of the divided graph. These results are then compared with the reference results of the original graph to highlight the impact of graph division on the robustness and resilience of the swarm. Namely, we analyze the performance of each algorithm on two aspects: the temporal evolution of the metrics, and the distribution of metrics values at three typical levels of network density (low, medium and high), as presented in Sec. 5.2.

5.3.1 Reference results

Fig. 4 shows the temporal evolution of each metric over a complete revolution: the blue curves represent the robustness metrics (flow robustness, routing cost and network efficiency), while red curves represent the resilience metrics (path redundancy, path disparity and node criticity).

Regarding robustness metrics, we observe that the flow robustness is overall equal to 1 (or 100%), except during a short period of time when it drops to 96%, disconnecting the graph. Nonetheless, the graph is connected the majority of the time. The routing cost slightly varies in time and is equal to 8170 transmissions in average, which means that the network needs to handle over 8000 voluminous data packet transmissions in order to complete an image computation. The network efficiency oscillates

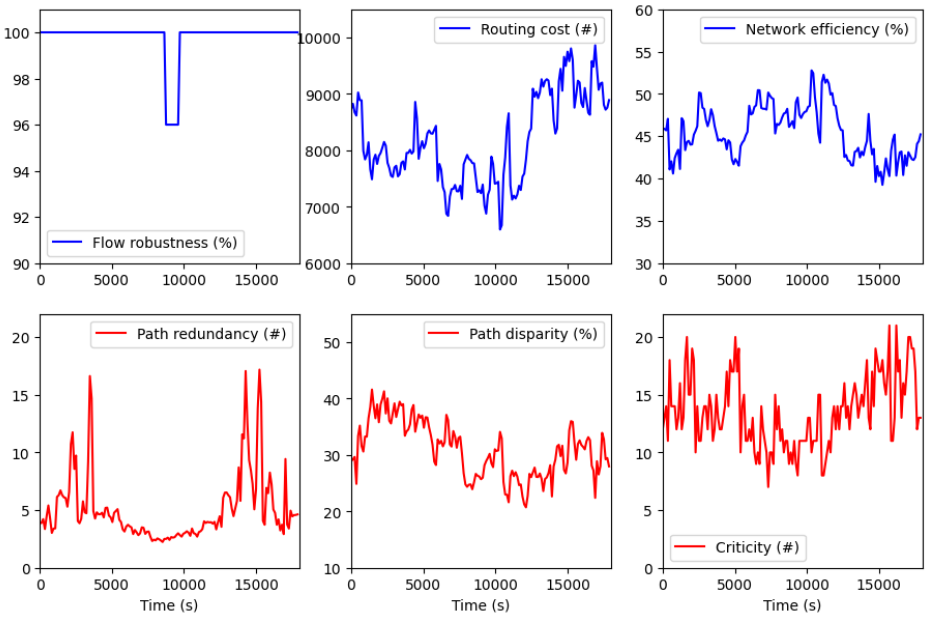


Fig. 4: Reference results on temporal evolution of robustness metrics (blue) and resilience metrics (red) on the undivided graph

around 45%, which can seem low but is not: from the definition by Eq. 5, the efficiency of a path of length 2 (*i.e.*, 2 hops between 3 vertices) is equal to 50% despite the path being particularly short, so an average network efficiency of 45% is fairly acceptable.

Regarding resilience metrics, we first notice that the path redundancy is very high at some times, reaching over 15 alternative paths between a pair of vertices. This level of redundancy is even more impressive because of our definition of alternative paths: in our case study, we define alternative paths as the set of all shortest paths between two vertices, and only the shortest paths. In other words, the path redundancy could be much higher if we decided to take into account slightly longer paths, but we do not deem it necessary in such graph. The disparity of these alternative paths is less impressive (around 30% in average), implying that the majority of the paths pass through the same vertices. It also indicates that a faulty node is very likely to affect multiple paths simultaneously. Finally, the node criticity is rather high: in average, 15 out of 50 nodes in the swarm are critical. This important criticity is a sign of low resilience and can be caused by the heterogeneous distribution of nanosatellites within the swarm.

In conclusion, the system shows a good level of robustness and is mostly able to withstand and avoid faults. However, the large path redundancy is compensated by poor disparity and high node criticity, which drastically lowers the level of resilience of the system, *i.e.*, its capacity to maintain functionality in the presence of faults. In these conditions, a reliability mechanism needs to be implemented to guarantee a good quality of service.

5.3.2 Results on robustness

Fig. 5 shows two types of figures: temporal curves in the left column, and boxplots in the right column. The temporal evolutions do not depict the values of the metrics at each timestamp, but rather the deviation of the metrics obtained after division by comparison to the reference metrics. The deviation can be positive, negative, or trivial in some cases. The figures in the right column show the distribution of each metric obtained with each graph partitioning (original, RND, MIRW, FFD and k-means) and for each level of network density (low, medium and high). The boxplots show the actual value of each metric, in opposition to the temporal deviation curves.

Fig. 5 depicts the results of robustness, which demonstrates that each division and clustering algorithm has a positive impact on robustness over time and at any network density level. The biggest gain is obtained on the routing cost (Fig. 5c), where each algorithm divides by almost ten the number of packets to transmit in the network (pay attention to the deviation scale). We can see in Fig. 5d that MIRW and k-means get the lowest routing cost independent of network density, in contrast to FFD, which gets the highest routing cost. This significant reduction in routing cost translates directly into longer mission durations and improved reliability in harsh environments. In either case, the reduction in packet transmission is drastic and has a very positive influence on the system's robustness. The flow robustness also increases after division, but very slightly as the graph is connected most of the time. K-means seems to mitigate the disconnection the best when it occurs (Fig. 5b). However, the gain is very trivial. Finally, regarding the network efficiency, Fig. 5e and Fig. 5f show that k-means and MIRW both stand out in performance, and improve the network efficiency from 45% to over 50%, which is an interesting improvement.

5.3.3 Results on resilience

The results on resilience are globally more mixed than those on robustness. Fig. 6 is arranged similarly to Fig. 5, with temporal curves on the left representing deviations, and boxplots on the right for distributions. This figure shows that none of the algorithms can improve all robustness and resilience metrics simultaneously.

The only improvement is performed by MIRW and k-means, and deals with the size of the critical set, as shown in Fig. 6e and Fig. 6f. We can clearly notice a reduction in the number of critical vertices with k-means, although MIRW performs poorly when the node density is low. On the other hand, RND and FFD tend to increase the number of critical vertices. Then, the results obtained for path redundancy are shown in Fig. 6a and Fig. 6b, proving that RND is the only algorithm that does not have a negative impact on resilience by not decreasing the redundancy of the original network. MIRW and k-means decrease the path redundancy level the most, which is to be expected as they perform the best at reducing the routing cost, and these two metrics are clearly antagonistic. A very similar observation is made on the path disparity presented in Fig. 6c and Fig. 6d. Despite the negative impact on resilience, it is important to highlight that MIRW and k-means decrease the redundancy level of an extremely redundant graph to begin with: depending on the application, such levels of path redundancy and disparity can be acceptable.

In conclusion, the algorithms MIRW and k-means algorithms clearly demonstrate better performance. K-means even outperforms MIRW in a few situations. However, in practice, an algorithm such as MIRW would be more interesting than k-means due to its computational complexity: indeed, MIRW is a distributed sequential algorithm with a very low complexity that scales easily. K-means, on the other hand, shows excellent results but comes at the cost of its complexity, which can cause problems when scaling large networks. Thus, MIRW is the best reliability mechanism to deploy in a nanosatellite swarm with limited computational power onboard.

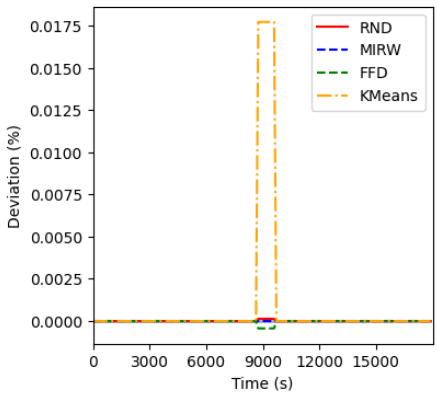
6 Conclusion and future research

Optimization of the reliability, *i.e.*, the robustness and resilience of a system, is crucial in Mobile Ad Hoc Networks (MANETs) and complex systems. A robust network can withstand internal and external faults, whereas a resilient network maintains functionality despite faults. This study focuses on enhancing the robustness and resilience of MANETs, which are treated as temporal graphs, through network architecture optimization. Our approach involves graph division by assigning each vertex to a group without isolating them. We examine the impact of random selection algorithms (RND), exploration algorithms (MIRW and FFD), and a clustering algorithm (k-means) on the original graph by analyzing flow robustness, routing cost, network efficiency, path redundancy, path disparity, and node criticity. Applying this evaluation protocol to the practical application of a nanosatellite swarm orbiting the Moon acting as a distributed radio-telescope, we demonstrate that division and clustering algorithms significantly improve the robustness of the system by reducing routing costs. Notably, MIRW and k-means outperform FFD and RND in reducing routing costs. MIRW and k-means overall excel in robustness optimization compared to the other algorithms. Regarding network resilience, MIRW and k-means decrease the number of critical vertices, but also decrease path redundancy and disparity, resulting in a mixed impact on the level of resilience. Overall, our results demonstrate that no algorithm is perfect for improving both robustness and resilience simultaneously, as some of their metrics are antagonistic. In conclusion, graph division and clustering techniques can drastically enhance the robustness of a system, but it comes at the cost of redundancy. Depending on the application and the original system, it may be problematic: in our study case, the original network is redundant enough to alleviate this issue and find an acceptable trade-off.

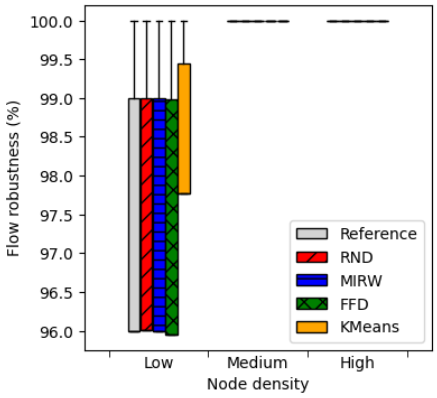
These findings not only enhance the theoretical understanding of MANETs but also provide actionable insights for real-world deployments in nanosatellite swarms.

Future research should explore the implementation of new algorithms derived from graph division and clustering techniques while considering their computational complexity (MIRW is lighter than k-means, but its performances could be improved). The search for new mechanisms that mitigate the robustness-resilience tradeoff is also crucial, as these two aspects should no longer be analyzed separately in exhaustive studies.

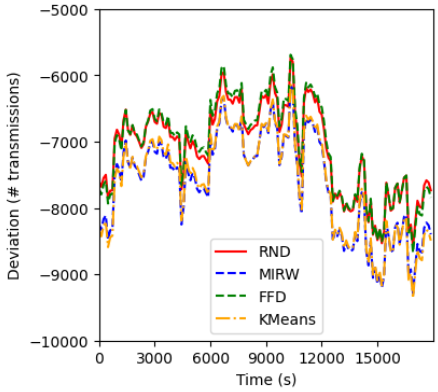
Acknowledgements. The authors would like to thank the CNES and the TéSA laboratory for supporting their work.



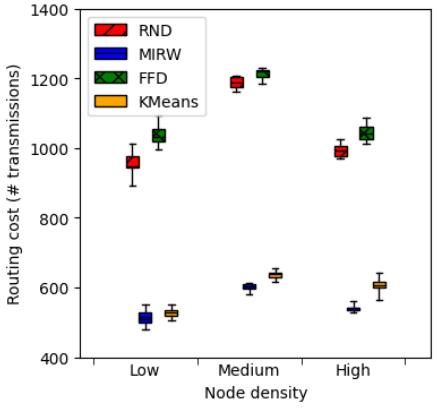
(a) Flow robustness (temporal)



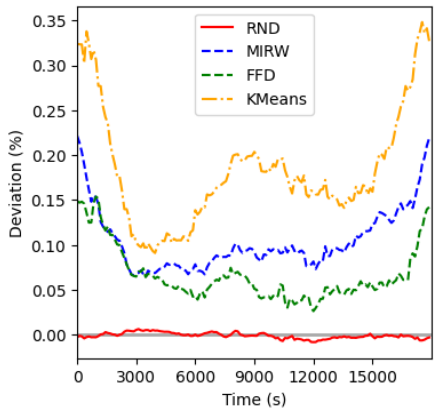
(b) Flow robustness (distribution)



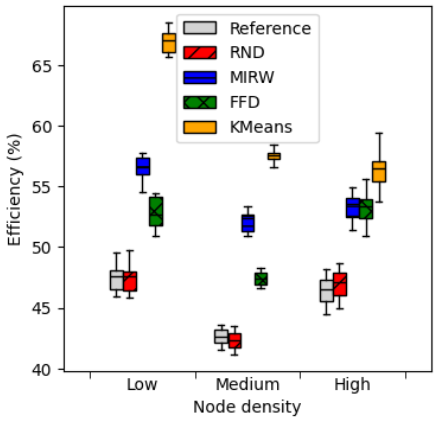
(c) Routing cost (temporal)



(d) Routing cost (distribution)

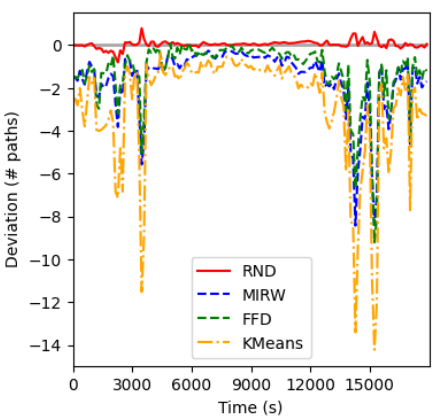


(e) Network efficiency (temporal)

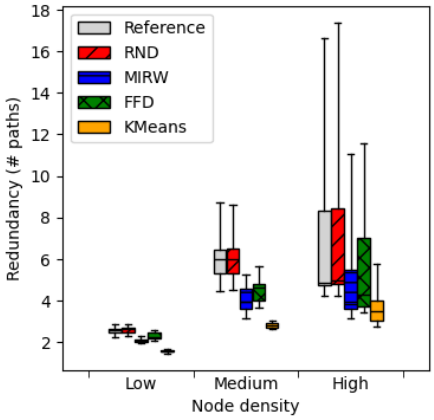


(f) Network efficiency (distribution)

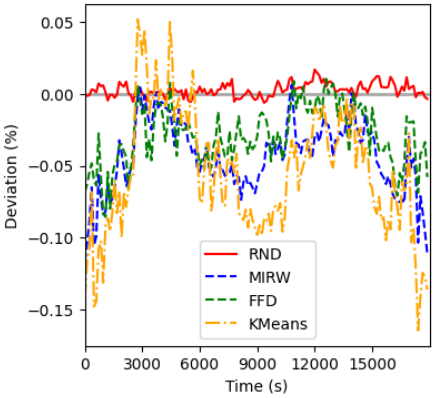
Fig. 5: Temporal evolution of robustness metrics (left) and their distributions at three levels of network density (right)



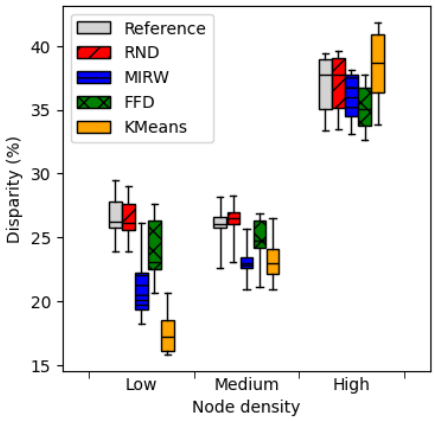
(a) Path redundancy (temporal)



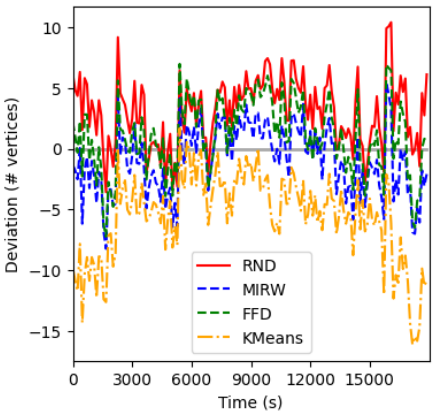
(b) Path redundancy (distribution)



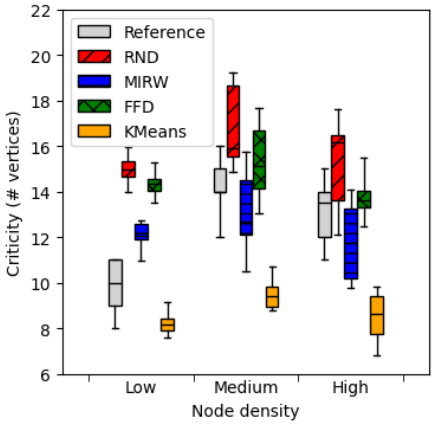
(c) Path disparity (temporal)



(d) Path disparity (distribution)



(e) Critical set size (temporal)



(f) Critical set size (distribution)

Fig. 6: Temporal evolution of resilience metrics (left) and their distributions at three levels of network density (right)

References

[1] Alyassi, R., Khonji, M., Karapetyan, A., Chau, S.C.-K., Elbassioni, K., Tseng, C.-M.: Autonomous recharging and flight mission planning for battery-operated autonomous drones. IEEE Transactions on Automation Science and Engineering **20 (2)**, 1034–1046 (2023)

[2] Rak, J., Hutchison, D. (eds.): Guide to Disaster-Resilient Communication Networks. Computer Communications and Networks. Springer, Cham, Switzerland (2020)

[3] Cecconi, B., Dekkali, M., Briand, C., al.: Noire study report: towards a low frequency radio interferometer in space. 2018 IEEE Aerospace Conference, 1–19 (2018)

[4] Akopyan, E., Dhaou, R., Lochin, E., Pontet, B., Sombrin, J.: Fair network division of nano-satellite swarms. 2023 IEEE 97th Vehicular Technology Conference (VTC2023-Spring), 1–5 (2023)

[5] Daanoune, I., Abdennaceur, B., Ballouk, A.: A comprehensive survey on leach-based clustering routing protocols in wireless sensor networks. Ad Hoc Networks **114**, 102409 (2021)

[6] Liu, W., Li, X., Liu, T., Liu, B.: Approximating betweeness centrality to identify key nodes in a weighted urban complex transportation network. Journal of Advanced Transportation (2019)

[7] Zhang, D., Cetinkaya, E.K., Sterbenz, J.P.: Robustness of mobile ad hoc networks under centrality-based attacks. 2013 5th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT), 229–235 (2013)

[8] Marydasan, B.P., Nadarajan, R.: Topology change aware on demand routing protocol for improving reliability and stability of manet. International journal of intelligent engineering and systems **15** (2022)

[9] Sun, W., Bocchini, P., Davidson, B.D.: Resilience metrics and measurement methods for transportation infrastructure: the state of the art. Sustainable and Resilient Infrastructure, 168–199 (2020)

[10] Sarkis, J., Zhu, Q.: Environmental sustainability and production: taking the road less travelled. International Journal of Production research, 743–759 (2018)

[11] Biggs, R., Schluter, M., Schoon, M.L.: Principles for Building Resilience: Sustaining Ecosystem Services in Social-ecological Systems. Cambridge University Press, Cambridge, United Kingdom (2015)

[12] Sterbenz, J.P.G., Hutchinson, D., Cetinkaya, E.K., Jabbar, A., Rohrer, J.P., Scholler, M., Smith, P.: Resilience and survivability in communication networks: Strategies, principles, and survey of disciplines. Computer Networks, 1245–1265 (2010)

[13] Kharrazi, A., Yu, Y., Jacob, A., Vora, N., Fath, B.D.: Redundancy, diversity, and modularity in network resilience: applications for international trade and implications for public policies. Current Research in Environmental Sustainability **2**, 100006 (2020)

[14] Ladas, A., Pavlatos, N., Weerasinghe, N., Politis, C.: Multipath routing approach to enhance resiliency and scalability in ad-hoc networks. 2016 IEEE International Conference on Communications (ICC), 1–6 (2016)

[15] Molla, D.M., Badis, H., Desta, A.A., George, L., Berbineau, M.: Sdr-based reliable and resilient wireless network for disaster rescue operations. 2019 International Conference on Information and Communication Technologies for Disaster Management (ICT-DM), 1–7 (2019)

[16] Jing, W., Xu, X., Pu, Y.: Route redundancy-based network topology measure of metro networks. Journal of Advanced Transportation **2019** (2019)

[17] Rajan, M.A., Chandra, M.G., Reddy, L.C., Hiremath, P.: Concepts of graph theory relevant to ad-hoc networks. International Journal of Computers, Communications and Control **3**, 465–469 (2008)

[18] Robinson, Y.H., Krishnan, R.S., Julie, E.G., Kumar, R., Son, L.H., Thong, P.H.: Neighbor knowledge-based rebroadcast algorithm for minimizing the routing overhead in mobile ad-hoc networks. Ad Hoc Networks **93**, 101896 (2019)

[19] Hu, P., Lau, W.C.: A survey and taxonomy of graph sampling. arXiv preprint arXiv :1308.5865 (2013)

[20] Akopyan, E.: Fiabilité de l'architecture réseau des systèmes spatiaux distribués sur essaims de nanosatellites. PhD thesis, Université de Toulouse (2024)

[21] Brown, L., Novaco, J.C.: Non thermal galactic emission below 10 mhz. The Astrophysical Journal (1978)

[22] Paimblanc, P., Mailhes, C.: Définition d'un référentiel spatio-temporel autonome dans une constellation de satellites. Technical report, CNES - TéSA (2018)